Lorrel Plimier **Project 1 Reflections – Pastry Chef** **November 14, 2018**

I really loved making this game. My goal with it (in addition to fulfilling the assignment requirements) was to have a functional game that my kids could play without breaking. And if they enjoyed it a little bit, then it would truly be a success.

I started by coming up with a pretty specific gameplay structure. I wrote this out for myself in plain English without worrying at first how I was going to code it. Because the assignment required classes, I had thought about that before submitting my design document. I knew that I wanted a Pastry_Chef player and rooms for the player to move through. I knew that I wanted desserts with specific qualities that I wanted the chef to bake. Those were the classes that I started with. As I developed the game, it became clear that some of my classes weren't what I thought they were and that I needed additional classes in order to make the game run smoothly. I also gave up fairly quickly on the idea of including multiple desserts. The game seemed complicated enough with just cakes, so I decided to stick with that, leaving cookies for another day.

One of my main struggles initially was understanding how to get the classes to communicate with each other. I could not understand how to get attributes to pass from class to class. I continually got "such and such" object has no attribute "such and such" errors. There was something that just didn't click for me about why I continued to get those errors. Taking some time away (sleeping on it) allowed my brain to figure it out and now I'm not even sure why I was confused. The second thing I struggled with was how to pass the gameplay in a way that didn't make my player go farther and farther down a rabbit hole. Before I figured this out, I would get error tracebacks that were extremely long! I knew I wanted some way to have the game-play continually come back to a central "location" in the code and then go back out again as the player made a new choice. Again, now that I understand this, I'm not sure why it was initially so confusing. Once I finally figured out this second roadblock, I felt like creating the game was pure fun. At that point, it was all a matter of homing in on the functionality that I wanted.

Once I finished creating all of the gameplay functionality, I could play the game myself and do everything that I wanted a player to be able to do within the game. But, my goal was for my kids to be able to play and I knew I wasn't there yet. One wrong keystroke at this stage would still spit them out of the game. So, my next phase of development was to make the code kid proof. The player had to be able to enter anything and attempt steps in any order without generating an error. I also wanted helpful messages to print on the screen, so I didn't have to sit with them and tell them what they should be doing. I wish I had written down how many lines of code I had before I starting adding hints and error handling, but I'm pretty sure that I doubled the length of my program. This was also where I really appreciated the use of classes because I could add a certain error handling to one function in a class and every other object that called that function would no longer generate an error. If it weren't for this, I probably would have tripled the length of my program! I've had both of my kids beta test the game and they found a few bugs for me, but luckily didn't generate any errors. And my ten-year-old even said, "This is kinda fun! I wouldn't want to play it all the time, but I like it." Yup… that's success in my book. My nine-year-old got a perfect score in delivering his first order (hard to do, in my opinion) and then looked at me as if to say, "are you going to make me do this again or can I go now?" Not surprising that my one, simple ascii animation wasn't enough to compete with his favorite game du jour – today it was Terraria.

To run the game, just execute the Pastry_Chef.py file. You should be able to easily play this game without knowing anything about it. Everything you need to know is given to you as you play (though you might lose some points along the way as you learn.) Some of the more entertaining text in the game is

through asking for help and trying to get orders in the “office” or “storefront.” One thing to note: this game works perfectly at my Windows command prompt. I haven’t tested it with Mac or Linux. The time delay (sleep function) and clear screen function do not work at my Git Bash prompt. This doesn’t affect the actual game play but it does affect the aesthetics of the game, causes time delays that are bunched together (you need to wait several seconds for text to appear at all,) and makes it less pleasing to read the text.

My kids recommended that I add an Easter egg to the game, so I have. It should be pretty easy to find if you look at the code (especially given my special “egg” module,) but you might not accidentally stumble upon it while playing the game. So, if you’d like to see it, keep taking eggs from the refrigerator. The clear screen and sleep functions must work properly in order to appreciate the Easter egg. It doesn’t work for me at the Git Bash prompt. The egg’s ascii animation is basically like a flip book. I’ve included the .txt file that has my individual animations, in case you want to take a look at them, since it is difficult to visualize them when they are in lists of strings with double back slashes.

Good luck and have fun!